

Beatrice Folino

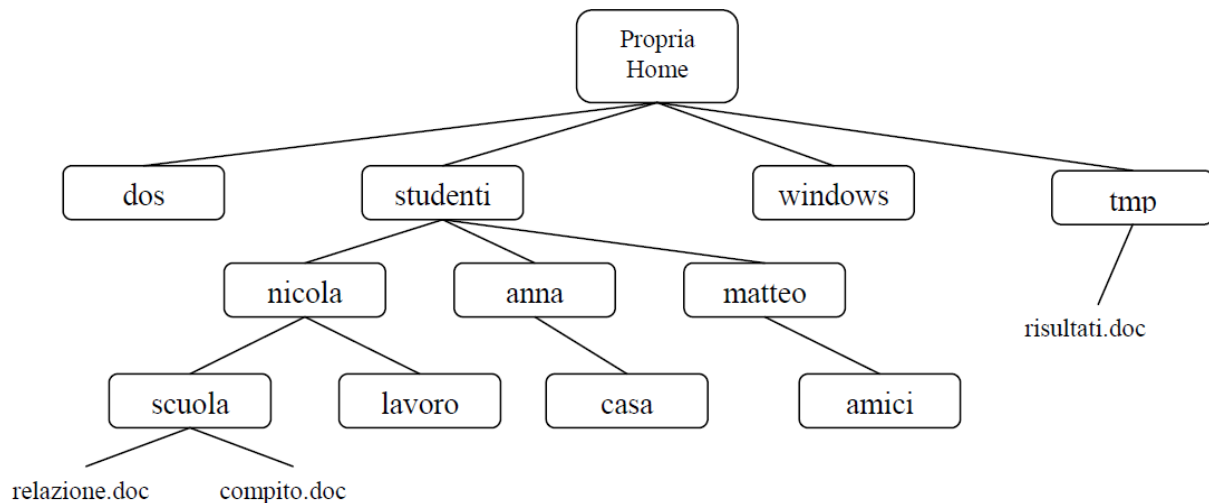
[Comandi su base Linux]

EPICODE - CYBERSECURITY CLASS [W5D1 Pratica1]

22 novembre 2023



Come prima cosa creare le seguenti cartelle e sottocartelle (usando i comandi “terminale” mkdir cd rmdir ... a partire dalla propria HOME e visualizzarle a video:



Ho a compimento la creazione delle cartelle come indicato:

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ ls  
Desktop Documents Downloads Music Pictures Public Templates Videos  
(kali@kali)-[~]  
$ mkdir dos  
(kali@kali)-[~]  
$ ls  
Desktop Documents dos Downloads Music Pictures Public Templates Videos  
(kali@kali)-[~]  
$ mkdir studenti  
(kali@kali)-[~]  
$ mkdir windows  
(kali@kali)-[~]  
$ mkdir tmp  
(kali@kali)-[~]  
$ ls  
Desktop Documents dos Downloads Music Pictures Public studenti Templates tmp Videos windows  
(kali@kali)-[~]  
$
```

L'immagine mostra uno screenshot del terminale di Kali Linux. Il prompt è "kali@kali: ~". Il menu di sistema mostra "File Actions Edit View Help". Le operazioni eseguite sono: "ls" (mostra Desktop, Documents, Downloads, Music, Pictures, Public, Templates, Videos), "mkdir dos", "ls" (mostra Desktop, Documents, dos, Downloads, Music, Pictures, Public, Templates, Videos), "mkdir studenti", "mkdir windows", "mkdir tmp", "ls" (mostra Desktop, Documents, dos, Downloads, Music, Pictures, Public, studenti, Templates, tmp, Videos, windows). Il prompt attuale è "\$".

PASSAGGIO DA UNA DIRECTORY A UN'ALTRA

Passaggio alla directory “casa” sotto “anna” in termini relativi e assoluti:

```
(kali㉿kali)-[~/studenti/matteo]
$ cd /home/kali/studenti

(kali㉿kali)-[~/studenti]
$ ls
anna matteo nicola

(kali㉿kali)-[~/studenti]
$ cd anna

(kali㉿kali)-[~/studenti/anna]
$ mkdir casa

(kali㉿kali)-[~/studenti/anna]
$ ls
casa
```

```
(kali㉿kali)-[~/studenti/nicola/lavoro]
$ cd /home/kali/studenti/anna/casa
```



a) COPIA DI UN FILE

Una volta che ci saremo spostati nella directory in cui è presente il file, la copia del file *compito.doc* avviene tramite il comando “cp”:

```
(kali㉿kali)-[~/studenti/nicola/scuola]
$ cp compito.doc /home/kali/studenti/anna/casa

(kali㉿kali)-[~/studenti/nicola/scuola]
$ cd /home/kali/studenti/anna/casa

(kali㉿kali)-[~/studenti/anna/casa]
$ ls
compito.doc
```

b) SPOSTARE UN FILE

Una volta che ci saremo spostati nella directory in cui è presente il file, potremo spostare *relazione.doc* tramite il comando “mv”:

```
(kali㉿kali)-[~/studenti/nicola/scuola]
$ mv relazione.doc /home/kali/studenti/anna/casa

(kali㉿kali)-[~/studenti/nicola/scuola]
$ ls
compito.doc

(kali㉿kali)-[~/studenti/nicola/scuola]
$ cd /home/kali/studenti/anna/casa

(kali㉿kali)-[~/studenti/anna/casa]
$ ls
compito.doc  relazione.doc
```

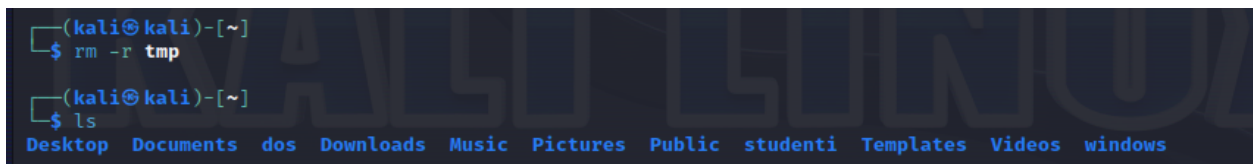


c) CANCELLARE UNA DIRECTORY

Per cancellare una directory su Linux, si può utilizzare il comando “rm” o “rmdir”. La scelta tra i due dipende dalla situazione: “rmdir” è utilizzato per cancellare directory vuote, mentre “rm” è utilizzato per cancellare directory che contengono file o sottodirectory.

Se la directory contiene ancora file o sottodirectory, “rmdir” restituirà un errore. In tal caso, si può utilizzare “rm” con l'opzione “-r” per cancellare ricorsivamente tutto il contenuto della directory. L'opzione “-r” sta per "ricorsivo" e permette di cancellare la directory insieme a tutti i suoi contenuti.

In questo caso dobbiamo eliminare la directory “tmp” che contiene il file *risultati.doc*:



```
(kali㉿kali)-[~]  
$ rm -r tmp  
  
(kali㉿kali)-[~]  
$ ls  
Desktop  Documents  dos  Downloads  Music  Pictures  Public  studenti  Templates  Videos  windows
```

La directory viene così eliminata correttamente.



d-e) CAMBIARE PERMESSI DI UN FILE

Possiamo utilizzare il comando **`chmod`** in Linux per cambiare gli attributi di un file. Per rendere un file scrivibile e leggibile solo per il proprietario e leggibile per gli altri, puoi usare la seguente sintassi:
chmod 644 nomefile

Qui, il numero "644" rappresenta i permessi del file. La rappresentazione numerica dei permessi funziona come segue:

- Il primo numero rappresenta i permessi del proprietario.
- Il secondo numero rappresenta i permessi del gruppo.
- Il terzo numero rappresenta i permessi per gli altri utenti.


I permessi sono assegnati in base alla somma dei valori seguenti:

- 4 per la lettura (r),
- 2 per la scrittura (w),
- 1 per l'esecuzione (x).

Quindi, nel caso di **`chmod 644`**:

- Il proprietario ha permessi di lettura e scrittura ($4 + 2 = 6$).
- Il gruppo e gli altri hanno solo permessi di lettura (4).

Questo garantirà che il proprietario possa leggere e scrivere il file, mentre gli altri possono solo leggerlo.



In questo caso la regola è stata applicata al file *pippo.txt* creato nella directory (con il comando **touch pippo.txt**):

```
(kali㉿kali)-[~]
$ ls
Desktop  Documents  dos  Downloads  Music  Pictures  pippo.txt  Public  studenti  Templates  Videos  windows

(kali㉿kali)-[~]
$ ls -l pippo.txt
-rw-rw-rw- 1 kali kali 0 Nov 22 05:09 pippo.txt

(kali㉿kali)-[~]
$ chmod 644 pippo.txt

(kali㉿kali)-[~]
$ ls -l pippo.txt
-rw-r--r-- 1 kali kali 0 Nov 22 05:09 pippo.txt
```

f) NASCONDERE UNA DIRECTORY

Per nascondere il contenuto di una directory in Linux, possiamo rinominarla in modo che il suo nome inizi con un punto ".". Le directory il cui nome inizia con un punto sono considerate nascoste in Linux.

Ecco un esempio di come puoi fare ciò utilizzando il comando **mv`:**

mv nomedirectory .nomedirectory

Dopo aver eseguito questo comando, il contenuto della directory sarà nascosto. Puoi ancora accedere alla directory, ma non sarà visibile quando elenchi i file nella directory principale senza l'opzione appropriata.

Per elencare anche le directory nascoste, puoi utilizzare l'opzione `-a` con il comando `ls`: **ls -a**

Questo mostrerà tutti i file, inclusi quelli il cui nome inizia con un punto.

In questo caso occorre nascondere contenuto della directory “anna”, per cui procediamo come da immagine seguente e poi verificare l’esito col comando **ls -a**:

```
(kali@kali)-[~]
$ cd /home/kali/studenti

(kali@kali)-[~/studenti]
$ ls
anna matteo nicola

(kali@kali)-[~/studenti]
$ mv anna .anna

(kali@kali)-[~/studenti]
$ ls
matteo nicola

(kali@kali)-[~/studenti]
$ ls -a
. .. .anna matteo nicola
```




g) VISUALIZZARE CONTENUTO FILE

Per visualizzare il contenuto di un file di testo in Linux, possiamo utilizzare il comando **cat** o **less** o **more** a seconda delle nostre preferenze.

1. **cat**: Il comando mostra il contenuto completo del file direttamente sul terminale. -> **cat nomefile**

2. **less**: Il comando permette di scorrere il contenuto del file una pagina alla volta. Possiamo uscire dalla visualizzazione premendo il tasto "q".
-> **less nomefile**

3. **more**: Funziona in modo simile a "less", consentendo di scorrere il contenuto del file pagina per pagina. -> **more nomefile**

Sostituiamo "nomefile" con il nome effettivo del nostro file di testo *pippo.txt* e usiamo per stavolta il comando **more**:

```
(kali㉿kali)-[~]  
$ more pippo.txt  
  
Questo file è stato creato per una esercitazione di Epicode.  
Il testo che viene visualizzato è puramente a scopo di esempio.
```

h) RIMUOVERE LA CARTELLA AMICI

Il comando assoluto usato in figura può essere eseguito da qualsiasi directory ci si trovi. La verifica può avvenire spostandosi nella directory e utilizzando **ls** per verificare che “matteo” non contiene più nulla.

```
(kali㉿kali)-[~/studenti]
$ rmdir /home/kali/studenti/matteo/amici

(kali㉿kali)-[~/studenti]
$ ls
matteo nicola

(kali㉿kali)-[~/studenti]
$ matteo

(kali㉿kali)-[~/studenti/matteo]
$ ls

(kali㉿kali)-[~/studenti/matteo]
$
```



h) RIMUOVERE TUTTE LE CARTELLE CREATE

Per comodità, mettendo caso di non ricordare quali directory contengono qualcosa, utilizzeremo sempre il comando **rm -r nomefile**

```
(kali㉿kali)-[~/studenti/matteo]
$ cd /home/kali

(kali㉿kali)-[~]
$ ls
Desktop  Documents  dos  Downloads  Music  Pictures  pippo.txt  Public  studenti  Templates  Videos  windows

(kali㉿kali)-[~]
$ rm -r dos

(kali㉿kali)-[~]
$ rm -r studenti

(kali㉿kali)-[~]
$ rm -r windows

(kali㉿kali)-[~]
$ rm -r pippo.txt

(kali㉿kali)-[~]
$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
```



PROVA DEI COMANDI

w -> Il comando `w` in Linux è utilizzato per visualizzare informazioni sulle utenze correntemente connesse al sistema, inclusi i dettagli come l'utente, l'orario di accesso, da dove si sono connessi e quali comandi stanno eseguendo.

who -> Il comando `who` in Linux è utilizzato per visualizzare informazioni sugli utenti attualmente connessi al sistema. La dicitura **tty7** che si vede in figura sta per “console virtuale”.

whoami -> Il comando `whoami` in Linux restituisce il nome dell'utente corrente associato al processo in esecuzione.

```
(kali㉿kali)-[~]
$ w
 05:26:44 up 50 min,  1 user,  load average: 0.27, 0.20, 0.13
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
kali      tty7     :0               04:36    50:39  21.66s  0.55s xfce4-session

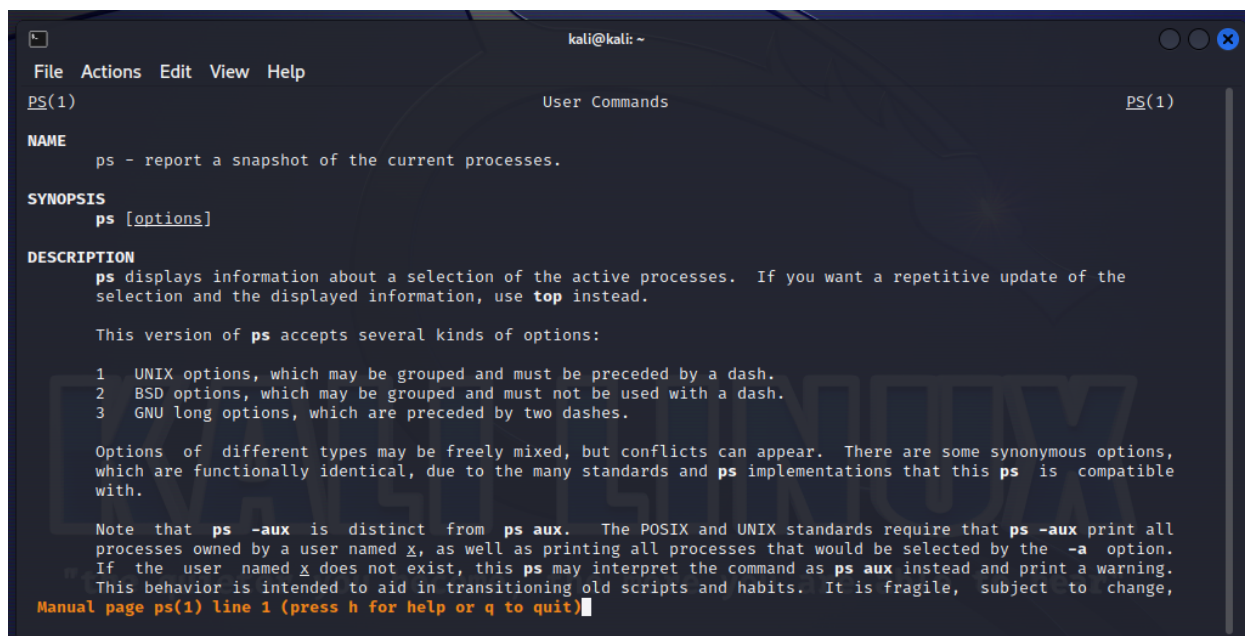
(kali㉿kali)-[~]
$ who
kali      tty7          2023-11-22 04:36 (:0)

(kali㉿kali)-[~]
$ whoami
kali
```

1-2) LEGGERE MANUALI DI jobs, pc E kill

Una volta aperto il terminale, e digitare il comando **man** seguito dall'azione che vogliamo scoprire per leggerne il manuale.

Così al comando **man ps** avremo la seguente risposta:



```
kali@kali: ~
File Actions Edit View Help
PS(1) User Commands PS(1)
NAME
  ps - report a snapshot of the current processes.
SYNOPSIS
  ps [options]
DESCRIPTION
  ps displays information about a selection of the active processes. If you want a repetitive update of the
  selection and the displayed information, use top instead.
  This version of ps accepts several kinds of options:
  1  UNIX options, which may be grouped and must be preceded by a dash.
  2  BSD options, which may be grouped and must not be used with a dash.
  3  GNU long options, which are preceded by two dashes.
  Options of different types may be freely mixed, but conflicts can appear. There are some synonymous options,
  which are functionally identical, due to the many standards and ps implementations that this ps is compatible
  with.
  Note that ps -aux is distinct from ps aux. The POSIX and UNIX standards require that ps -aux print all
  processes owned by a user named x, as well as printing all processes that would be selected by the -a option.
  If the user named x does not exist, this ps may interpret the command as ps aux instead and print a warning.
  This behavior is intended to aid in transitioning old scripts and habits. It is fragile, subject to change,
  Manual page ps(1) line 1 (press h for help or q to quit)
```

Mentre per il comando **man kill** avremo:

```
kali@kali: ~  
File Actions Edit View Help  
KILL(1) User Commands KILL(1)  
NAME  
kill - send a signal to a process  
SYNOPSIS  
kill [options] <pid> [...]  
DESCRIPTION  
The default signal for kill is TERM. Use -l or -L to list available signals. Particularly useful signals include HUP, INT, KILL, STOP, CONT, and 0. Alternate signals may be specified in three ways: -9, -SIGKILL or -KILL. Negative PID values may be used to choose whole process groups; see the PGID column in ps command output. A PID of -1 is special; it indicates all processes except the kill process itself and init.  
OPTIONS  
<pid> [...]  
    Send signal to every <pid> listed.  
-<signal>  
-s <signal>  
--signal <signal>  
    Specify the signal to be sent. The signal can be specified by using name or number. The behavior of signals is explained in signal(7) manual page.  
-q, --queue value  
    Use sigqueue(3) rather than kill(2) and the value argument is used to specify an integer to be sent with  
Manual page kill(1) line 1 (press h for help or q to quit)
```

Il comando **jobs**, invece, non ha manuale richiamabile direttamente, ma occorre digitare **man builtins** che ci permetterà di visualizzare tutta la banca dati dei manuali presenti e scorrere fino a quando non troviamo quello relativo a **jobs**:

```
jobs [-lnprs] [ jobspec ... ]  
jobs -x command [ args ... ]  
The first form lists the active jobs. The options have the following meanings:  
-l    List process IDs in addition to the normal information.  
-n    Display information only about jobs that have changed status since the user was last notified of their status.  
-p    List only the process ID of the job's process group leader.  
-r    Display only running jobs.  
-s    Display only stopped jobs.  
  
If jobspec is given, output is restricted to information about that job. The return status is 0 unless an invalid option is encountered or an invalid jobspec is supplied.  
  
If the -x option is supplied, jobs replaces any jobspec found in command or args with the corresponding process group ID, and executes command passing it args, returning its exit status.
```



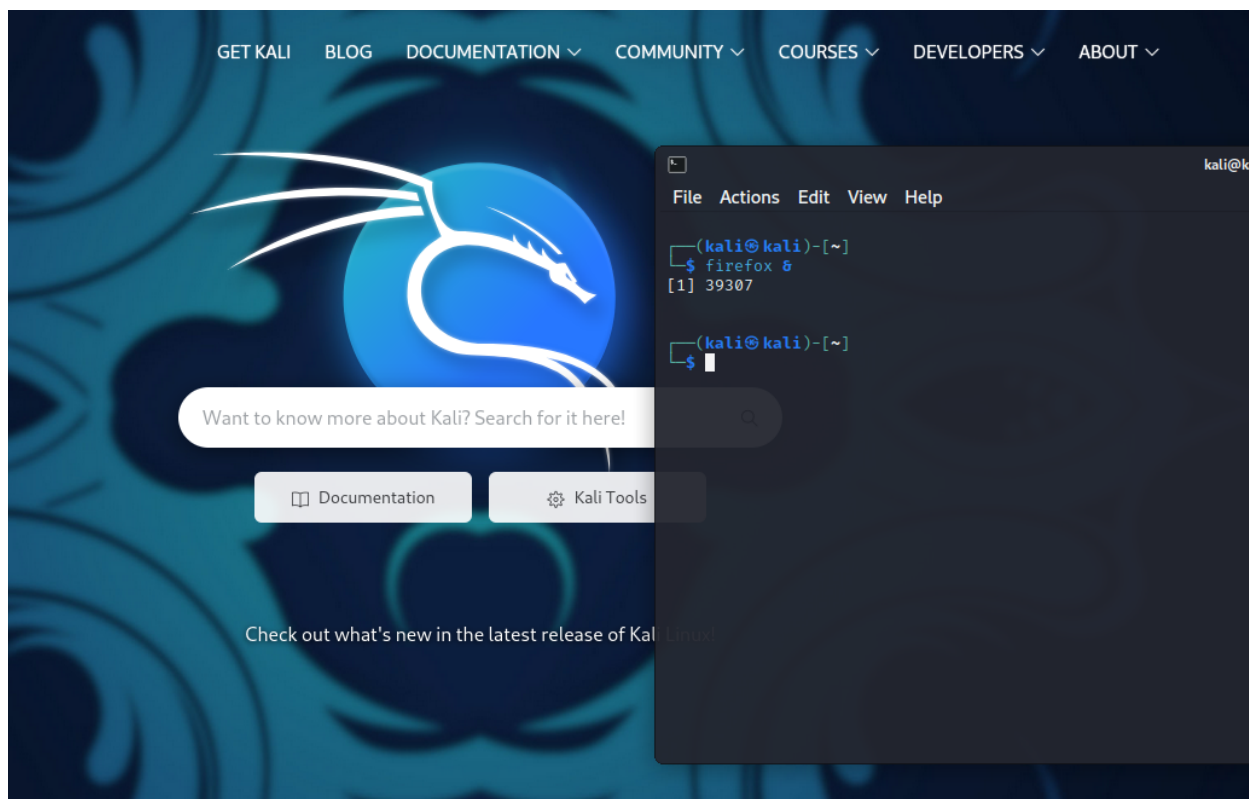
Digitando **kill PID** (dove PID è il codice assegnato al processo identificato) il processo scelto verrà interrotto. In questo caso intendiamo bloccare il processo iniziato digitando **vi pippo**. Se tutto sarà stato svolto correttamente, avremo in risposta la finestra di stop da parte del terminale.

14

6-7-8) AVVIO DI FIREFOX IN BACKGROUND E TERMINAZIONE DEL PROCESSO

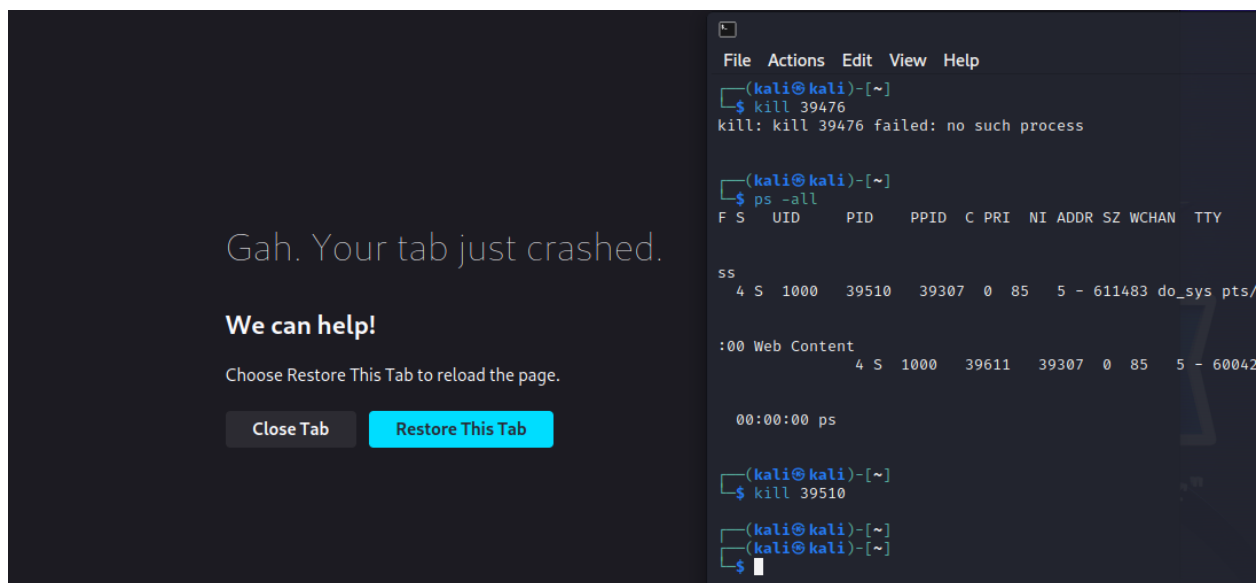
Per lanciare il comando firefox in background su Linux, possiamo utilizzare l'operatore **&** alla fine -> **firefox &**

Questo comando avvia Firefox in background, consentendo di continuare a utilizzare il terminale senza aspettare che Firefox termini l'esecuzione.



Per terminare il processo è sufficiente utilizzare il comando **kill PID**, dove PID va sostituito col codice numerico del processo corrispondente a Firefox (ricordiamo che il codice può essere recuperato col comando **ps -all**).

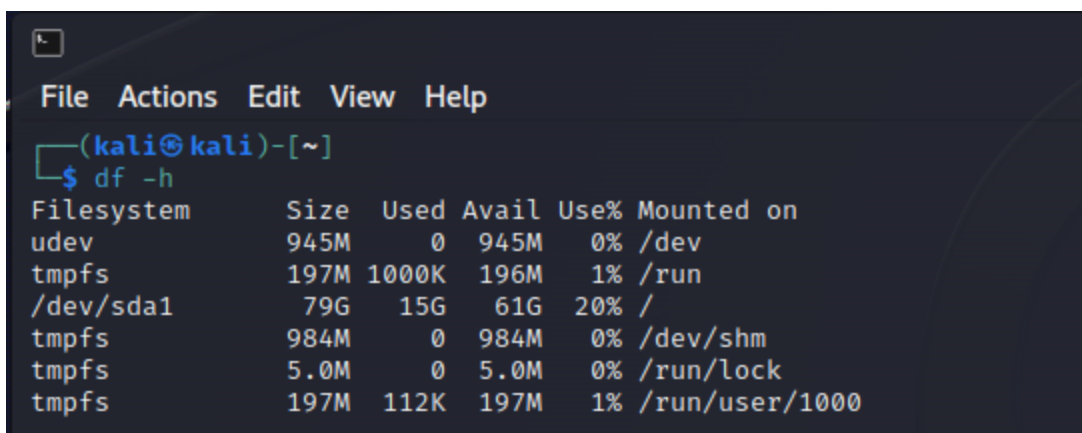
Se tutto si sarà svolto correttamente, avremo in risposta la schermata di crash dell'applicazione.



9) VERIFICA SPAZIO SU DISCO

Per verificare quanto spazio è occupato su disco in Linux, puoi utilizzare il comando `df` (disk free).

Un esempio è il comando **`df -h`**, questo comando mostra una panoramica delle partizioni del disco nel tuo sistema, inclusi dettagli come lo spazio totale, lo spazio utilizzato, lo spazio libero e il punto di montaggio. L'opzione **`-h`** indica di visualizzare i risultati in un formato più leggibile per l'utente, utilizzando unità di misura come KB, MB, GB, ecc.



```
(kali㉿kali)-[~]
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            945M     0  945M   0% /dev
tmpfs           197M 1000K   196M   1% /run
/dev/sda1        79G   15G   61G  20% /
tmpfs           984M     0  984M   0% /dev/shm
tmpfs           5.0M     0   5.0M   0% /run/lock
tmpfs           197M  112K   197M   1% /run/user/1000
```